destination site address field(s) **418**. The fixed fields **410** also include rule fields **420**, security fields **422**, fixed software agent executable code fields **424** and fixed data fields **426**. The variable fields **450** include: a new/old flag **452**, rule fields **454**, result fields **456**, variable software agent executable code fields **458** and variable data fields **460**.

The security fields **422** typically comprise information/protocols for enhancing security of the origin site, the trusted site, the destination site, the agent (e.g., the new/old flag) or any combination thereof. As an example, the security fields can provide for lock and key protocol(s), with or without encryption. Whatever security protocol is implemented, the origin site, the trusted site and the destination site have resident information/code site, as respectively appropriate, for effecting the security protocol (e.g., for unlocking and decrypting an agent or part thereof).

Although the invention has been disclosed with reference to several embodiments, it is to be understood that, from reading this description, those of skill in the art may appreciate changes and modification that may be made which do not depart from the scope and spirit of the invention as described above and claimed hereafter.

We claim:

1. A computing environment comprising:

an origin site capable of creating and sending a software agent;

a destination site capable of receiving, executing and returning a software agent;

a trusted site having software implemented thereon for receiving a software agent from an origin site, fingerprinting the software agent, forwarding the agent to the destination site, comparing original and return software agent fingerprints, and sending verification notices to the origin site; and

an open communication network, where the sites are in communication with each other through the network.

2. The environment of claim **1**, wherein the network is an Internet.

3. The environment of claim **1**, wherein the network operates using World Wide Web protocols.

4. The environment of claim **1**, wherein the fingerprint is a one-way hash function.

5. The environment of claim **1**, wherein the software agent comprises a plurality of fixed fields and a plurality of variable fields and wherein the fingerprint comprises information taken substantially from the fixed fields.

6. The environment of claim **5**, wherein the fixed fields include a set of rules, the set of rules enabling the trusted site to check that variable information has been appropriately modified at the destination site.

7. The method of claim **6**, wherein the fingerprint is a one-way hash function.

8. A method implemented on a trusted site of a distributed computing environment, the method comprising the steps of:

receiving a software agent from an origin site;

generating an original agent fingerprint of the agent;

sending the agent to a destination site;

receiving the agent returning from the destination site;

generating an return agent fingerprint of the agent;

comparing the original agent fingerprint to a return agent fingerprint; and

sending a verification return to the origin site.

9. The method of claim **8**, further comprising the steps of:

generating an agent return timer;

monitoring for the agent's return;

sending a time out notification to the origin site if the agent return timer indicates a time-out condition;

executing agent rules designed to inform the trusted site on the proper handling of variable agent information;

checking the variable agent information against the rules upon the agent's return from its destination site; and

sending a notification to the origin site so that the origin site can determine whether the variable information was modified appropriately.

10. The method of claim **8**, wherein the environment includes an open communication network.

11. The method of claim **10**, wherein the communication network is an Internet.

12. A method implemented on a trusted site of a distributed computing environment comprising the steps of:

receiving a software agent from a origin site;

generating an original agent fingerprint;

generating an agent return timer;

sending the agent to a destination site;

monitoring for the agent's return;

sending a time out notification to the origin site if the agent return timer indicates a time-out condition;

generating a return agent fingerprint if the agent returns from the destination site prior to the agent return timer timing out;

comparing the original and return agent fingerprints to form a verification result; and

sending verification result to the origin site.

13. The method of claim **12**, wherein the fingerprint is a one-way hash function.

14. The method of claim **12**, further comprising the step of:

logging the activities set forth in the other steps.

15. The method of claim **12**, wherein the environment includes an open communication network.

16. The method of claim **15**, wherein the communication network is an Internet.

17. A software agent implemented in a memory of a trusted site comprising:

a set of routing fields;

a set of fixed rules field to be executed by the trusted site when the agent returns to the trusted site from a destination site;

a new agent flag field so that the trusted site can identify the agent as a returning agent;

a fixed code field including fixed data and executable code which is designed to remain unmodified during task execution; and

a variable code field including variable data and variable executable code which is designed to be modified during task execution.

18. The agent of claim **17**, further comprising:

a security field for identifying the user site and for locking out other specified information.

19. The agent of claim **17**, wherein the set of rules are designed to determine whether a given agent task has executed within a predetermined manner by comparing rule-based information to information in the variable fields when the agent returns to the trusted site.

20. The agent of claim **19**, the rule-based information includes acceptable ranges of values for each data variable associated with the agent.

* * * * *